



PATENT ABSTRACTS OF JAPAN

(11) Publication number: 08297564 A

(43) Date of publication of application: 12.11.1996

(51) Int. Cl. G06F 5/00
H03M 7/40

(21) Application number: 07101353
(22) Date of filing: 25.04.1995

(71) Applicant: FUJI XEROX CO LTD
(72) Inventor: ITO ATSUSHI

(54) DATA COMPRESSION DEVICE

COPYRIGHT: (C)1996,JPO

(57) Abstract:

PURPOSE: To obtain a high compressibility by properly encoding a data sequence having plural data sequence characteristics.

CONSTITUTION: A data counting part 2 accumulates the appearance frequency of each character in the held data sequence to generate a histogram. Further, a dictionary counting part 7 accumulates the appearance frequency of each character to dictionary data in a dictionary part 6 as a result of encoding to generate a histogram. A comparison part 3 compares the histogram generated by the data counting part 2 with the histogram generated by the dictionary counting part 7 and supplies the coincidence degree to a signal generation part 4. The signal generation part 4 sends a dictionary reset control signal S1 out when it is judged that data characteristics change. A reset universal encoding part 5 performs universal encoding for the data held by the data holding part 1 thereafter while generating new dictionary data.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-297564

(43)公開日 平成8年(1996)11月12日

(51)Int.Cl. ⁶	識別記号	序内整理番号	F I	技術表示箇所
G 0 6 F 5/00			G 0 6 F 5/00	H
H 0 3 M 7/40		9382-5K	H 0 3 M 7/40	

審査請求 未請求 請求項の数4 O L (全 17 頁)

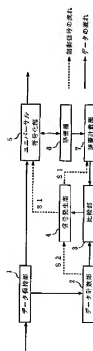
(21)出願番号	特願平7-101353	(71)出願人	000005496 富士ゼロックス株式会社 東京都港区赤坂二丁目17番22号
(22)出願日	平成7年(1995)4月25日	(72)発明者	伊藤 篤 神奈川県海老名市本郷2274番地 富士ゼロックス株式会社内
		(74)代理人	弁理士 川▲崎▼ 研二 (外1名)

(54)【発明の名称】 データ圧縮装置

(57)【要約】

【目的】 複数のデータ列特性が混在するデータ列に対しても、適切な符号化を実行し、高い圧縮率を実現する。

【構成】 データ計数部2は、保持されたデータ列の各文字の出現数を累積し、そのヒストグラムを作成する。また、辞書計数部7は、符号化実行によって辞書部6に作成された辞書データに対して各文字の出現数を累積し、そのヒストグラムを作成する。比較部3は、データ計数部2によるヒストグラムと辞書計数部7によるヒストグラムとを比較し、その一致度を信号発生部4へ供給する。信号発生部4は、比較結果に基づいて、データ特性が変化したと判断された場合には辞書リセット制御信号S1を送出する。リセットされたユニバーサル符号化部5は、それ以降、データ保持部1に保持されたデータに対して、新たな辞書データを作成しながらユニバーサル符号化を行う。



【特許請求の範囲】

【請求項1】 過去に現れたデータ列の情報を参照して、以後に現れるデータ列を符号化するデータ圧縮装置において、

入力されるデータ列の特性変化を検出するデータ特性変化検出手段と、

符号化されたデータ列とこれに対する符号とが登録される辞書手段と、

前記辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを前記辞書手段に登録する符号化手段と、

前記データ特性変化検出手段による検出結果に基づいて、前記辞書内容および前記符号化手段の動作をリセットする辞書リセット制御手段とを具備することを特徴とするデータ圧縮装置。

【請求項2】 過去に現れたデータ列の情報を参照して、以後に現れるデータ列を符号化するデータ圧縮装置において、

入力される符号化前のデータ列における文字出現数を累計するデータ計数手段と、

符号化されたデータ列とこれに対する符号とが登録される辞書手段と、

前記辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを前記辞書手段に登録する符号化手段と、

前記辞書手段に記憶され、前記符号化手段によって既に符号化されたデータ列における文字出現数を累計する辞書計数手段と、

前記データ計数手段による累計値と前記辞書計数手段による累計値とを比較する比較手段と、

前記比較手段による比較結果に基づいて、前記辞書内容および前記符号化手段の動作をリセットする辞書リセット制御手段とを具備することを特徴とするデータ圧縮装置。

【請求項3】 過去に現れたデータ列の情報を参照して、以後に現れるデータ列を符号化するデータ圧縮装置において、

入力される符号化前のデータ列における文字出現数を累計するデータ計数手段と、

前記データ計数手段による過去の累計値を記憶する履歴計数手段と、

符号化されたデータ列とこれに対する符号とが登録される辞書手段と、

前記辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを前記辞書手段に登録する符号化手段と、

前記データ計数手段による累計値と前記履歴計数手段に記憶された累計値とを比較する比較手段と、

前記比較手段による比較結果に基づいて、前記辞書内容および前記符号化手段の動作をリセットする辞書リセッ

ト制御手段とを具備することを特徴とするデータ圧縮装置。

【請求項4】 過去に現れたデータ列の情報を参照して、以後に現れるデータ列を符号化するデータ圧縮装置において、

入力される符号化前の隣接するデータ列における文字出現数を累計するデータ計数手段と、

符号化されたデータ列とこれに対する符号とが登録される辞書手段と、

前記辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを前記辞書手段に登録する符号化手段と、

前記データ計数手段による隣接するデータ列における累計値の変化を抽出する抽出手段と、

前記抽出手段による抽出結果に基づいて、前記辞書内容および前記符号化手段の動作をリセットする辞書リセット制御手段とを具備することを特徴とするデータ圧縮装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、過去に現れたデータ列の部分データ列の情報を参照し、以後に現れるデータ列を符号化するデータ圧縮装置に関する。

【0002】

【従来の技術】 近年、文字コードや画像データをコンピュータ、計算機などで扱うようになり、また、取り扱うデータ量が增大しつつある。したがって、大量データを記憶媒体に蓄積したり、転送を行う場合に効率的な圧縮を用いてデータ量を削減することが必須となっている。

【0003】 従来より、データ圧縮装置による圧縮方式としては、ユニバーサル符号化による方式がある。ユニバーサル符号は、可逆型（情報保存型）のデータ圧縮復元方式であり、圧縮時に出現する情報源の統計的性質に

よらずに適用することができる。すなわち、さまざまな種類のデータには、それぞれの特徴があり、その特徴を用いて冗長性を削減することによりデータ圧縮を行う。

【0004】 以下の説明においては、データの1ワード単位を文字と呼び、文字が任意の個数だけ連なったものを文字列と呼ぶことにする。

【0005】 ユニバーサル符号の1つとしてレンペルジブ（Lempel-Ziv）方式がある。このレンペルジブ方式には、スライド辞書法（ユニバーサル型）と動的辞書法（増分分解型）があり、スライド辞書法アルゴリズムとしては、LZSS方式（T.A.Bell, "Better OMP/L Text Compression", IEEE Trans. on Comm., Vol. COM-34, No. 12, Dec. 1986）があり、動的辞書法アルゴリズムとしては、LZW方式（T.A.Welch, "A Technique for High Performance Data Compression", IEEE Computer, vol. 17, June 1984）がある。また、動的辞書法には、辞書リセット機能を付随した方式（特開平1-132222号）

がある。

【0005】(1) LZW方式

LZW方式は、動的辞書法を基盤としており、過去に現れたデータ列の統計的性質を反映させながら、符号化を行う方式である。LZW方式の構成・アルゴリズムとしては、書き換え可能な辞書を設け、過去に出現したデータ列の部分データ列を参考に上記辞書に辞書登録を行い、現在、符号化を行うデータ列に対して、上記辞書から最大一致系列を探索し、その番号を符号として符号化を行う。

【0006】A. LZW方式による符号化

まず、LZW方式による符号化について説明する。ここで、Kを文字(1データ)、 ω を語頭文字列(prefix string)、または文字列登録番号とする。また、説明を簡単にするために、a, b, cの3文字を情報源とする文字列に対する圧縮手順を例として述べる。

【0007】図10は、LZW方式による符号化の手順を示すフローチャートであり、図11は、辞書登録テーブルの一例を示す概念図、図12は、LZW方式による符号化の過程を示す概念図である。まず、ステップS1において、予め、全文字列に対して文字分だけ辞書に初期登録を行う。ここでは、a, b, cのそれぞれ一文字の文字列を初期登録することになる。

【0008】次に、ステップS2において、最初の文字Kを入力し、これを語頭文字列 ω とする。ステップS3では、次の入力データが存在するか否かを判断する。そして、次の入力データが存在する場合には、ステップS3における判断結果は「YES」となり、ステップS4へ進む。ステップS4では、次の文字Kを入力する。次に、ステップS5において、辞書に部分文字列 ωK が存在するか否かを判断する。部分文字列 ωK とは、ステップS2で入力した一文字 ω とステップS4で入力した文字Kからなる文字列である。

【0009】そして、辞書に部分文字列 ωK が存在すれば、ステップS5における判断結果は「YES」となり、ステップS6へ進む。ステップS6では、部分文字列 ωK を語頭文字列 ω と改め、ステップS3へ戻る。以下、ステップS3～S6を繰り返し実行することで、現在、符号化を試みている文字列に対して、辞書内の最長一致系列を探索する。つまり、入力文字を一文字ずつ取り込み、その文字列が辞書に登録されているか否かを判別しているわけである。

【0010】そして、ステップS4で文字Kを入力した結果、部分文字列 ωK が辞書に存在しない場合には、ステップS5における判断結果は「NO」となり、ステップS7へ進む。ステップS7では、符号として、それまで入力した部分文字列 ω に対する文字列登録番号code(ω)を出力し、かつ辞書に部分文字列 ωK を登録するとともに、文字Kを語頭文字列 ω とし、ステップS3へ戻る。以下、ステップS3～S7を繰り返し実行し、

次の文字列に対して新たに辞書内を探索し、文字列番号code(ω)の出力と、辞書への登録とを行う。

【0011】そして、次の入力データが存在しない場合には、ステップS3における判断結果は「NO」となり、ステップS4へ進む。ステップS4では、語頭文字列(文字列登録番号)code(ω)を出力し、当該処理を終了する。

【0012】次に、上述したLZW方式による符号化の具体例を図11および図12を参照して説明する。まず、a, b, cのそれぞれ一文字を辞書に初期登録する。この段階で、文字aには文字列登録番号として「1」が設定され、文字bには「2」、文字cには「3」が設定される。次に、最初の文字K=aを入力し、aに対する文字列登録番号code(ω)=「1」とする。次の文字K=bを入力し、部分文字列 ωK =「ab」が辞書に存在するか否かを探索する。ここで、実際には、部分文字列 ωK は、「語頭文字列」+「拡張文字」、すなわち「1b」と表される。

【0013】最長一致系列探索により部分文字列「ab」が辞書に存在しないので、符号として、文字列登録番号code(ω)=「1」が出力される。さらに、部分文字列 ωK =「1b」、すなわち部分文字列「ab」が登録番号「4」として辞書に登録される。また、語頭文字列 ω が文字「b」に更新される。

【0014】次に、文字K=「a」を入力し、部分文字列 ωK =「ba」、すなわち「2a」について最長一致系列探索を行う。この場合、辞書には、部分文字列「b a」が存在しないので、符号として、文字列登録番号code(ω)=「2」が出力される。さらに、部分文字列 ωK =「2a」が登録番号「5」として辞書に登録され、語頭文字列 ω が文字「a」に更新される。

【0015】さらに、次の文字K=「b」を入力する。この場合、部分文字列 ωK =「ab」、すなわち「1b」は辞書に存在するので、該部分文字列「1b」の文字列登録番号code(ω)=「4」を語頭文字列 ω とする。次に、文字K=「c」を入力し、部分文字列 ωK =「abc」、すなわち「4c」について最長一致系列探索を行う。この場合、辞書には、部分文字列「abc」=「4c」が存在しないので、符号として文字列登録番号code(ω)=「4」が出力される。さらに、部分文字列 ωK =「4c」が登録番号「6」として辞書に登録される。

【0016】以下、同様にして、最長一致系列探索・符号化・辞書登録を繰り返し行うことで、LZW符号化が実行される。

【0017】B. LZW方式による復号化

次に、LZW方式による復号化について説明する。図13は、LZW方式による復号化の手順を示すフローチャートであり、図14は、LZW方式による復号化の過程を示す概念図である。まず、符号化同様、ステップS1

0において、全文字に対して、一文字分だけ辞書に初期登録を行う。次に、ステップS11において、最初の一文字CODEを入力し、該符号CODEを符号OLDcodeに代入する。最初の一文字は、登録文字列のうち、どれかの文字列に対応するので、入力符号CODEに対応する文字Kを出力し、該文字KをバッファFINcharへ格納する。

【0018】次に、ステップS12において、次の符号が存在するか否かを判断する。そして、次の符号が存在する場合には、ステップS12における判断結果は「YES」となり、ステップS13へ進む。ステップS13では、次の符号CODEを入力し、該符号CODEを符号INcodeへ代入する。次に、ステップS14において、上記符号CODEが定義されているか否かを判断する。そして、定義されていれば、ステップS14における判断結果は「NO」となり、そのままステップS16へ進む。

【0019】一方、定義されていなければ、ステップS14における判断結果は「YES」となり、ステップS15へ進む。ステップS15では、バッファFINcharから文字を出力し、符号OLDcodeを符号CODEに代入し、部分文字列「符号OLDcode+文字FINchar」に対する新たな文字列登録番号を符号INcodeに代入する。その後、ステップS16へ進む。

【0020】ステップS16では、文字列のKが辞書に存在するか否かを判断する。文字列のKが辞書に存在すれば、ステップS16における判断結果は「YES」となり、ステップS17へ進む。ステップS17では、文字列のKの文字KをPUSHstackに代入し、語頭文字列のωを符号CODEに代入する。そして、ステップS16へ戻る。

【0021】一方、文字列のKが辞書に存在しなければ、すなわち、文字列が文字Kであった場合には、ステップS16における判断結果は「NO」となり、ステップS18へ進む。ステップS18では、文字Kを出力するとともに、文字KをバッファFINcharへ代入する。次に、ステップS19において、PUSHstackが空になるまで、PUSHstackの先頭(top)文字を出力し、POPstackを行う。さらに、ステップS20において、部分文字列「符号OLDcode+文字K」を辞書に登録し、符号INcodeを符号OLDcodeに代入して、ステップS12へ戻る。以下、入力符号がなくなるまで、上述したステップS12～S20を繰り返して実行し、符号化を行う。そして、入力符号がなくなると、ステップS12における判断結果が「NO」となり、当該処理を終了する。

【0022】次に、上述したLZW方式による復号化の具体例を図14を参照して説明する。まず、全一文字を初期登録する。次に、最初の一文字「1」を入力し、符号OLDcodeに符号「1」を代入して、辞書内の符号

「1」に対応する文字「a」を出力するとともに、該文字「a」をバッファFINcharへ格納しておく。そして、次の符号「2」を入力すると同時に、該符号「2」を符号INcodeに代入しておく。ここで、該符号「2」が定義されていて、しかも文字列が文字「b」のみなので、該文字bを出力するとともに、該文字bをバッファFINcharへ格納しておく。

【0023】ここで、スタックstackは、空なので、部分文字列「1b」を登録番号「4」として辞書登録し、符号INcode(=「2」)を符号OLDcodeに格納する。符号「4」は辞書に存在し、しかも符号「4」=ωKとなるような「ω」が存在するので(ここでは「K」は文字「b」)、文字「b」をスタックへプッシュし、符号CODEを符号code(ω)に置き換える。このとき、語頭文字列ωは、文字「a」であるので、文字「a」を出力し、また、スタックから文字「b」を出力し、さらに、部分文字列「ba」を辞書登録した後に、次の符号を入力する。このような処理を入力符号がなくなるまで繰り返して実行することで、LZW方式による復号化が行われる。

【0024】

【発明が解決しようとする課題】しかしながら、上述した従来のデータ圧縮装置による符号化方式では、異なった傾向を示す複数の文字列が混在したデータに対して、動的辞書法を用いて圧縮処理を行う場合には、圧縮効率が悪いという問題があった。例えば、LZW方式では、今までとは傾向の異なるデータ列が入力されると、その時点までに作成された辞書登録テーブルを用いて符号化を行うため、適切な符号化処理が実行されず、圧縮率の低下をもたらすという問題があった。

【0025】また、辞書リセット機能付き動的辞書法(特開平1-132222号)では、入力データ・ストリームと出力データ・ストリームに対応したセグメントのビット長を比較し、上記セグメントに対する圧縮率が低下すると、辞書を初期化状態にする、すなわち辞書リセットを行う。しかしながら、この方式では、上記セグメントに対して圧縮率の測定を行い、その結果に基づいて辞書リセットを行うので、辞書リセットのタイミングが遅れてしまい、上記セグメントの分だけ入力データに対してふさわしくない辞書登録テーブルを用いることになる。このため、上記セグメントの部分の圧縮率が低下するだけでなく、セグメントの部分で適切な辞書登録を行うことができず、以後に入力されるデータ列に対して符号化効率に悪影響を及ぼし、全データに対する圧縮率の低下が復号化速度を低下させてしまうという問題があった。

【0026】この発明は上述した事情に鑑みてなされたもので、複数のデータ列特性が混在するデータ列や、データ列特性変化部を持つデータ列に対しても、適切な辞

書を作成することができるとともに、適切な符号化を実行することができ、高圧縮率を得ることができるデータ圧縮装置を提供することを目的としている。

【0027】

【課題を解決するための手段】上述した問題点を解決するために、請求項1記載の発明では、過去に現れたデータ列の情報を参照して、以後に現れるデータ列を符号化するデータ圧縮装置において、入力されるデータ列の特性変化を検出するデータ特性変化検出手段と、符号化されたデータ列とこれに対する符号とが登録される辞書手段と、前記辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを前記辞書手段に登録する符号化手段と、前記データ特性変化検出手段による検出結果に基づいて、前記辞書内容および前記符号化手段の動作をリセットする辞書リセット制御手段とを具備することを特徴とする。

【0028】また、請求項2記載の発明では、過去に現れたデータ列の情報を参照して、以後に現れるデータ列を符号化するデータ圧縮装置において、入力される符号化前のデータ列における文字出現数を累計するデータ計数手段と、符号化されたデータ列とこれに対する符号とが登録される辞書手段と、前記辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを前記辞書手段に登録する符号化手段と、前記辞書手段に記憶され、前記符号化手段によって既に符号化されたデータ列における文字出現数を累計する辞書計数手段と、前記データ計数手段による累計値と前記辞書計数手段による累計値とを比較する比較手段と、前記比較手段による比較結果に基づいて、前記辞書内容および前記符号化手段の動作をリセットする辞書リセット制御手段とを具備することを特徴とする。

【0029】また、請求項3記載の発明では、過去に現れたデータ列の情報を参照して、以後に現れるデータ列を符号化するデータ圧縮装置において、入力される符号化前のデータ列における文字出現数を累計するデータ計数手段と、前記データ計数手段による過去の累計値を記憶する履歴計数手段と、符号化されたデータ列とこれに対する符号とが登録される辞書手段と、前記辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを前記辞書手段に登録する符号化手段と、前記データ計数手段による累計値と前記履歴計数手段に記憶された累計値とを比較する比較手段と、前記比較手段による比較結果に基づいて、前記辞書内容および前記符号化手段の動作をリセットする辞書リセット制御手段とを具備することを特徴とする。

【0030】また、請求項4記載の発明では、過去に現れたデータ列の情報を参照して、以後に現れるデータ列を符号化するデータ圧縮装置において、入力される符号化前の隣接するデータ列における文字出現数を累計する

データ計数手段と、符号化されたデータ列とこれに対する符号とが登録される辞書手段と、前記辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを前記辞書手段に登録する符号化手段と、前記データ計数手段による隣接するデータ列における累計値の変化を抽出する抽出手段と、前記抽出手段による抽出結果に基づいて、前記辞書内容および前記符号化手段の動作をリセットする辞書リセット制御手段とを具備することを特徴とする。

【0031】

【作用】請求項1記載の発明によれば、符号化手段は、辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを辞書手段に登録する。この符号化過程において、データ特性変化検出手段は、入力される符号化前のデータ列の特性変化を検出する。辞書リセット制御手段は、データ特性変化検出手段による検出結果に基づいて、データ列のデータ特性に変化があった場合には、辞書内容および符号化手段の動作をリセットする。この結果、符号化手段は、新たな辞書データを作成しながら符号化を続行する。したがって、複数のデータ列特性が混在するデータ列や、データ列特性変化部を持つデータ列に対しても、適切な辞書を作成することが可能になるとともに、適切な符号化を実行することができ、高圧縮率を得ることが可能となる。

【0032】請求項2記載の発明によれば、符号化手段は、辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを辞書手段に登録する。この符号化過程において、データ計数手段は、入力される符号化前のデータ列における文字出現数を累計する。また、辞書計数手段は、辞書手段に記憶され、符号化手段によって既に符号化されたデータ列における文字出現数を累計する。また、比較手段は、データ計数手段による累計値と辞書計数手段による累計値とを比較する。そして、辞書リセット制御手段は、比較手段による比較結果に基づいて、データ列のデータ特性に変化があった場合には、辞書内容および符号化手段の動作をリセットする。この結果、符号化手段は、新たな辞書データを作成しながら符号化を続行する。したがって、複数のデータ列特性が混在するデータ列や、データ列特性変化部を持つデータ列に対しても、適切な辞書を作成することが可能になるとともに、適切な符号化を実行することができ、高圧縮率を得ることが可能となる。

【0033】請求項3記載の発明によれば、符号化手段は、辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを辞書手段に登録する。この符号化過程において、データ計数手段は、入力される符号化前のデータ列における文字出現数を累計する。履歴計数手段には、データ計数

手段による過去の累計値が記憶される。また、比較手段は、データ計数手段による累計値と履歴計数手段に記憶された累計値とを比較する。そして、辞書リセット制御手段は、比較手段による比較結果に基づいて、データ列のデータ特性に変化があった場合には、辞書内容および符号化手段の動作をリセットする。この結果、符号化手段は、新たな辞書データを作成しながら符号化を続行する。したがって、複数のデータ列特性が混在するデータ列や、データ列特性変化部を持つデータ列に対しても、適切な辞書を作成することが可能になるとともに、適切な符号化を実行することができ、高圧縮率を得ることが可能となる。

【0034】請求項4記載の発明によれば、符号化手段は、辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを辞書手段に登録する。この符号化過程において、データ計数手段は、入力される符号化前のデータ列における文字出現数を累計する。抽出手段は、データ計数手段による隣接するデータ列における累計値の変化を抽出する。そして、辞書リセット制御手段は、抽出手段による抽出結果に基づいて、データ列のデータ特性に変化があった場合には、辞書内容および符号化手段の動作をリセットする。この結果、符号化手段は、新たな辞書データを作成しながら符号化を続行する。したがって、複数のデータ列特性が混在するデータ列や、データ列特性変化部を持つデータ列に対しても、適切な辞書を作成することが可能になるとともに、適切な符号化を実行することができ、高圧縮率を得ることが可能となる。

【0035】

【実施例】次に図面を参照してこの発明の一実施例について説明する。

A. 第1の実施例

A-1. 第1の実施例によるデータ圧縮・伸張装置の構成

図1は、本発明の第1の実施例によるデータ圧縮・伸張装置の符号化部の構成を示すブロック図である。図において、符号化部は、データ保持部1、データ計数部2、比較部3、信号発生部4、ユニバーサル符号化部5、辞書部6、および辞書計数部7から構成されている。データ保持部1は、入力したデータを一時保持し、所定量ずつデータ計数部2とユニバーサル符号化部5へ供給する。データ計数部2は、データカウント機能を備えており、データ保持部1に保持され、ユニバーサル符号化部5で符号化される前のデータ列に対して、各文字（1文字または文字列）の出現数を計数し、そのヒストグラムを作成し、比較部3へ供給する。また、辞書リセット（初期化）後の初期値は、辞書リセットの実行を回避する辞書リセット回避信号S2を信号発生部4に送出する機能を備えている。

【0036】比較部3は、上記データ計数部2からの計

数結果と、後述する辞書データ計数部7からの計数結果に対して、所定のしきい値（図3参照）を与え、該しきい値を越えるデータ値を抽出し、該抽出部分の一致度を算出し、該算出結果を信号発生部4へ供給する。信号発生部4は、比較部3の算出結果に基づいて、すなわち、データ特性が変化しと判断された場合には、ユニバーサル符号化部5、辞書計数部7に辞書リセット制御信号S1を送出する。

【0037】ユニバーサル符号化部5は、データ保持部1から所定量ずつ供給されるデータ列に対して、辞書部6の情報に基づいて、ユニバーサル符号化を施して圧縮した後、出力する。なお、ユニバーサル符号化部5へ供給されるデータ列は、既に、データ計数部2によって計数され、比較部3でデータ特性変化が判断されたものである。また、ユニバーサル符号化部5は、上記辞書リセット制御信号S1が供給されると、それまでの辞書部6の情報を初期化し、新たに辞書部6の情報を作成しながらユニバーサル符号化を実施する。

【0038】また、上述した辞書リセット制御信号S1は、辞書リセットを実行するか否かを知らせるための信号であり、辞書リセットを実行しない場合であっても、ユニバーサル符号化部5に送信される。辞書リセット制御信号S1により辞書リセットを実行しない場合には、辞書リセット制御信号S1を受信したユニバーサル符号化部5は、次の入力データをデータ保持部1から入力し、符号化を実行する。また、辞書リセット制御信号S1により辞書リセットを実行する場合には、辞書リセット制御信号S1を受信したユニバーサル符号化部5は、辞書リセット制御信号を出力し、辞書をリセットし、その後、次の入力データをデータ保持部1から入力し、符号化を実行する。

【0039】また、上記辞書部6の情報は、前述した図11に示すものと同じであり、ユニバーサル符号化部5による符号化処理も、辞書リセット制御信号S1に対する処理（後述）を除いて、従来の符号化方式と同一である。また、辞書計数部7は、符号化実行によって辞書部6に作成された辞書を参照して、辞書データに対して、各文字（1文字または文字列）の出現数を累積し、そのヒストグラムを作成し、前述した比較部3へ供給する。

【0040】また、図2は、当該データ圧縮・伸張装置の復号化部の構成を示すブロック図である。図において、復号化部は、ユニバーサル復号化部8と、辞書部6とから構成されている。ユニバーサル復号化部8は、辞書リセット制御信号を検知する機能を備えている。この辞書リセット制御信号検知機能は、復号化しようとする入力符号を入力した時点での辞書部6内の辞書登録番号のうち、最大登録番号nに対して、 $\log_2 n$ ビットの符号か、もしくは $(\log_2 n) + 1$ ビットの符号が入力された場合に、辞書リセットを実行するというものである。

【0041】辞書リセットを実行する場合には、ユニバーサル符号化部5が出力する辞書リセット制御信号は、辞書リセットを実行する直前までに、辞書部6に登録された登録番号のうち、最大登録番号1によって決定される。実際に出力される辞書リセット制御信号は、 $[log_2 n]$ ビットの符号か、もしくは $[log_2 n] + 1$ ビットの符号である。また、 $[log_2 n]$ は、 $log_2 n$ を越えない最大の整数であるとする。

【0042】A-2、第1の実施例の動作
次に、上述した第1の実施例の動作について説明する。

(1) 符号化

まず、辞書をリセットし、続いて、入力データをデータ保持部1を介してユニバーサル符号化部5に所定のデータ量だけ入力し、ユニバーサル符号化を実行する。ここで、次に符号化すべきデータはデータ保持部1に保持され、この保持データに対してはデータ計数部2によってデータ特性判別が行われる。データ計数部2は、図4に示すように、データ列（データ・ストリーム）の中から、特定の一部分（セグメント）に対して、各文字（1文字または文字列）の出現数を計数する。セグメントは、処理の進捗状況に応じて、図示の 슬라이ド 幅毎にずらして行く。該 슬라이ド 幅は任意に設定可能である。

【0043】具体的には、データ計数部2は、データ保持部1から読み込んだべき所定量のデータ（例えば、図4に示すセグメント1）に対して、各文字（1文字または文字列）の出現数を計数し、そのヒストグラムを作成する。その結果が例えは図3に示すラインBのようになったとする。また、辞書計数部7は、符号化実行によって辞書部6に作成された辞書を参照して、辞書データに対して、各文字（1文字または文字列）の出現数を累積し、そのヒストグラムを作成する。その結果が例えは図3に示すラインAのようになったとする。辞書部6に作成された辞書は、過去における符号化で作成されたものであるため、その辞書データは、これまでの符号化におけるデータ列のデータ特性を示している。

【0044】比較部3は、所定のしきい値（図3参照）を与え、ラインA（データ特性検出結果）とラインB（辞書データ計数結果）のそれぞれに対し、上記しきい値を越えるデータ値を抽出し、該抽出部分の一致度を算出し、この算出結果を信号発生部4へ供給する。なお、図3に示すラインAとラインBとは、通常、各々のデータの絶対数が異なるので、比較可能にするために、適当な尺度を用いて比率分布表を作成し、該比率分布表を用いて比較するようにしてもよい。また、該比率分布表による比較は、後述する第2の実施例、第3の実施例においても実行可能である。ここで、一致度が高い場合には、データ保持部1に保持され、これから符号化されるデータ列のデータ特性が、既に符号化されたデータ列のデータ特性と近似しているということになる。一方、一致度が低い場合には、データ保持部1に保持され、これ

から符号化されるデータ列のデータ特性が、既に符号化されたデータ列のデータ特性とは異なるものであるということになる。

【0045】信号発生部4は、上述した比較部3の算出結果に基づいて、データ特性が変化しと判断された場合には、ユニバーサル符号化部5、辞書計数部7に辞書リセット制御信号S1を送出する。ここで、辞書計数部7が辞書リセット制御信号S1によりリセット（初期化）される。この後、データ保持部1に保持しておいたデータをユニバーサル符号化部5に入力し、新たに辞書データを作成しながらユニバーサル符号化を行う。

【0046】ここで、ユニバーサル符号化部5の動作を図5を参照して説明する。なお、図において、ステップS30～S36は、図10に示すステップS1～S7に対応し、ステップS39は、図10に示すステップS8に対応する。これら対応する部分については簡単に説明する。まず、ステップS30において、予め、全文字列に対して一文字分だけ辞書に初期登録を行った後、ステップS31において、最初の一文字Kを入力し、これを語頭文字列 ω とする。次に、ステップS32では、次の入力データが存在するかどうかを判断し、次の入力データが存在する場合には、ステップS33へ進み、次の文字Kを入力する。そして、ステップS34において、辞書に部分文字列のKが存在するかどうかを判断し、辞書に部分文字列のKが存在すれば、ステップS35へ進み、ステップS35では、部分文字列のKを語頭文字列 ω と改め、ステップS32へ戻る。以下、ステップS32～S35を繰り返して実行することで、現在、符号化を試みている文字列に対して、辞書内の最長一致系列を探索する。

【0047】そして、ステップS33で文字Kを入力した結果、部分文字列 ω のKが辞書に存在しない場合には、ステップS34からステップS36へ進み、符号として、それまで入力した部分文字列 ω に対する文字列登録番号code(ω)を出力し、かつ辞書に部分文字列のKを登録するとともに、文字Kを語頭文字列 ω とし、ステップS37へ進む。ステップS37では、辞書リセット制御信号S1が供給されたかどうかを判断する。そして、辞書リセット制御信号S1が供給されていないければ、ステップS37における判断結果は「NO」となり、ステップS32へ戻る。以下、ステップS32～S37を繰り返して実行し、次の文字列に対して新たに辞書内を探索し、文字列番号code(ω)の出力と、辞書への登録とを行う。

【0048】一方、辞書リセット制御信号S1が供給された場合には、ステップS38へ進み、辞書リセット（初期化）した後、ステップS30へ戻る。以下、データ保持部1に保持しておいたデータ（特性変化の生じたセグメント）をユニバーサル符号化部5に入力し、新たに、上述したステップS30の初期登録からステップS

38を繰り返して実行し、ユニバーサル符号化を行う。

【0049】一方、次の入力データが存在しない場合には、ステップS32における判断結果は「NO」となり、ステップS39へ進み、語頭文字列（文字列登録番号）code (ω) を出力し、当該処理を終了する。

【0050】(2) 復号化

次に、復号化においては、ユニバーサル復号化部8は、復号化しようとする入力符号が入力された時点での辞書9内の辞書登録番号（図11を参照）のうち、最大登録番号nに対して、 $[10g_2n]$ ビットの符号か、もしくは $[10g_2n] + 1$ ビットの符号が入力された場合には、辞書リセットを実行する。その後、ユニバーサル復号化部8は、再び、入力される符号に対して復号化の実行を継続する。

【0051】ここで、ユニバーサル復号化部8の動作を図6を参照して説明する。なお、図において、ステップS40～S43は、図13に示すステップS10～S13に対応し、ステップS45～S47は、図10に示すステップS14～S20に対応する。これらに対応する部分については簡単に説明する。まず、符号化同様、ステップS40において、全文字に対して、一文字分だけ辞書に初期登録を行い、ステップS41において、最初の符号CODEを入力し、該符号CODEを符号OLDecodeに代入し、入力符号CODEに対応する文字Kを出力し、該文字KをバッファF INcharに格納する。

【0052】次に、ステップS42において、次の符号が存在するか否かを判断し、次の符号が存在する場合には、ステップS43へ進み、次の符号CODEを入力し、該符号CODEを符号INcodeへ代入する。次に、ステップS44において、リセットコードが供給されたか否かを判断する。そして、リセットコードが供給されていない場合には、ステップS44における判断結果は「NO」となり、ステップS45へ進む。ステップS45では、上記符号CODEが定義されているか否かを判断し、定義されていれば、そのままステップS47へ進む。

【0053】一方、定義されていないければ、ステップS45からステップS46へ進み、バッファF INcharから文字を出力し、符号OLDecodeを符号CODEに代入し、部分文字列「符号OLDecode+文字F INchar」に対する新たな文字列登録番号を符号INcodeに代入する。その後、ステップS47へ進む。

【0054】ステップS47では、文字列ωのKが辞書に存在するか否かを判断し、文字列ωのKが辞書に存在すれば、ステップS48へ進み、文字列ωの文字KをPUSHstackに代入し、語頭文字列ωを符号CODEに代入する。そして、ステップS47へ戻る。

【0055】一方、文字列ωのKが辞書に存在しなければ、すなわち、文字列が文字Kであった場合には、ステップS47からステップS49へ進み、文字Kを出力す

るとともに、文字KをバッファF INcharへ代入する。次に、ステップS50において、PUSHstackが空になるまで、PUSHstackの先頭(top)文字を出力し、POPstackを行う。さらに、ステップS51において、部分文字列「符号OLDecode+文字K」を辞書に登録し、符号INcodeを符号OLDecodeに代入して、ステップS42へ戻る。以下、入力符号がなくなるまで、上述したステップS42～S51を繰り返して実行し、符号化を行う。そして、入力符号がなくなると、ステップS42における判断結果が「NO」となり、当該処理を終了する。

【0056】一方、リセットコードが供給された場合には、ステップS44における判断結果が「YES」となり、そのままステップS40へ戻る。その後、ユニバーサル復号化部8は、ステップS40～S51を繰り返して実行し、再び、入力される符号に対して新たな復号化の実行を継続する。

【0057】B. 第2の実施例

B-1. 第2の実施例によるデータ圧縮・伸張装置の構成

図7は、本発明の第2の実施例によるデータ圧縮・伸張装置の構成を示すブロック図である。なお、図1に対応する部分については同一の符号を付けて説明を省略する。図において、履歴計数部10は、データ計数部2によって算出された各文字の出現数から得たヒストグラムを格納し、そのヒストグラムを比較部11へ供給する。また、履歴計数部10は、辞書リセット制御信号S1が供給されると、履歴情報（過去のヒストグラム）をリセット（初期化）する。比較部11は、データ計数部2から供給される、次に符号化されるデータ列のヒストグラムと、履歴データ計数部10から供給される、過去のデータ列のヒストグラムとを比較し、この比較結果を信号発生部4へ供給する。

【0058】B-2. 第2の実施例の動作

次に、上述した第2の実施例の動作について説明する。

(1) 符号化

まず、動作の前提として、履歴計数部10には、辞書リセットが行われた時に、ユニバーサル符号化が限りに実行された、所定のデータ量のデータに対して、例えば、前回のリセットから今回のリセットまでのデータに対して、データ計数部2で作成されたヒストグラム（例えば、図3に示すラインA）が格納される。符号化に際しては、まず、データ計数部2において、データ保持部1から読み込んできた所定のデータ量に対して、第1の実施例と同様に、ヒストグラムを作成する。この作成結果が例えば図3に示すラインBのようになったとする。

【0059】次に、比較部11は、所定のしきい値（図3参照）を与え、ラインA（履歴）とラインBのそれぞれに対して、しきい値を越えるデータ値を抽出し、抽出部分の一致度を算出し、この算出結果を信号発生部4へ

供給する。信号発生部4は、データ特性が変化しと判断された場合には、ユニバーサル符号化部5、履歴計数部10に辞書リセット制御信号S1を送出する。ここで、履歴計数部10は、辞書リセット制御信号S1により、履歴情報（過去のヒストグラム）をリセット（初期化）する。この後、データ保持部1に保持しておいたデータをユニバーサル符号化部5に入力し、新たに辞書データを作成しながらユニバーサル符号化を行う。一方、ユニバーサル符号化部5は、データ特性が変化していない場合には、従来のデータ符号化と同じ処理を実行する。

【0060】（2）復号化

図2に示すユニバーサル復号化部8は、復号化しようとする入力符号が入力された時点での辞書9内の辞書登録番号のうち、最大登録番号nに対して、 $[log_2 n]$ ビットの符号か、もしくは $[log_2 n] + 1$ ビットの符号が入力された場合に、辞書リセットを実行する。その後、ユニバーサル復号化部8は、再び、入力される符号に対して新たな復号化の実行を継続する。

【0061】C. 第3の実施例

C-1. 第3の実施例によるデータ圧縮・伸張装置の構成

図8は、本発明の第3の実施例によるデータ圧縮・伸張装置の構成を示すブロック図である。なお、図1または図7に対応する部分については同一の符号を付けて説明を省略する。図において、データ計数部12は、データカウント機能を備えており、データ保持部1から読み込んだ、連続した隣接するセグメントの各データ列に対して、各文字（1文字または文字列）の出現数を計数し、そのヒストグラムを作成し、履歴計数部10および抽出部13へ供給する。例えば、図9に示すように、連続した2つのセグメント1、2のデータ列①、②に対して、順次、各文字（1文字または文字列）の出現数を計数する。また、辞書リセット（初期化）後のある期間は、辞書リセットの実行を回避する辞書リセット回避信号S2を信号発生部14に送出する機能を備えている。

【0062】履歴計数部10には、データ計数部12で計数された最初のセグメントのヒストグラムが格納される。このヒストグラムは、上記データ計数部12が後半のセグメントのヒストグラムを作成し、抽出部13へ供給した時点で、抽出部13へ供給される。抽出部13は、履歴計数部10から供給される1つ前のデータ列（例えば、データ列①）に対するヒストグラム（パターン）と、データ計数部12から供給される次のデータ列（例えば、データ列②）に対するヒストグラム（パターン）とを比較し、抽出部分の一致度を算出し、この算出結果を信号発生部14へ供給する。

【0063】C-2. 第3の実施例の動作

次に、上述した第3の実施例の動作について説明する。

（1）符号化

まず、動作の前提として、履歴計数部10には、辞書リセットが行われた時に、ユニバーサル符号化が既に実行された、過去のデータ列に対して、データ計数部12で作成されたヒストグラム（例えば、図3に示すラインA）が格納される。符号化に際しては、データ計数部12において、データ保持部1から読み込んだ、図9に示す2つの隣接したセグメント1、2のデータ列①、②に対して、ヒストグラムを作成する。この作成結果が、各々、例えば図3に示すラインA、ラインBのようになったとする。

【0064】抽出部13は、所定のしきい値（図3参照）を与え、ラインAとラインBのそれぞれに対して、しきい値を越えるデータ値を抽出し、抽出部分の一致度を算出し、この算出結果を信号発生部14へ供給する。信号発生部14は、抽出部13の算出結果に基づいて、データ特性が変化しと判断した場合には、ユニバーサル符号化部5に辞書リセット制御信号S1を送出する。ここで、ユニバーサル符号化部5が辞書リセット制御信号S1によりリセット（初期化）される。この後、データ保持部に保持しておいたデータをユニバーサル符号化部5に入力し、ユニバーサル符号化を行う。一方、ユニバーサル符号化部5は、データ特性が変化していない場合には、従来のデータ符号化と同じ処理を実行する。

【0065】（2）復号化

図2に示すユニバーサル復号化部8は、復号化しようとする入力符号が入力された時点での辞書9内の辞書登録番号のうち、最大登録番号nに対して、 $[log_2 n]$ ビットの符号か、もしくは $[log_2 n] + 1$ ビットの符号が入力された場合に、辞書リセットを実行する。その後、ユニバーサル復号化部8は、再び、入力される符号に対して新たな復号化の実行を継続する。

【0066】なお、上述した第1の実施例におけるデータ計数部2、比較部3、および辞書計数部7、または、第2の実施例におけるデータ計数部2、履歴計数部10、および比較部11、さらに、第3の実施例における履歴計数部10、データ計数部12、抽出部13は、データ特性の変化を検出するデータ特性判別部を構成し、対象となるデータ列の各文字（1文字または文字列）の出現数を計数し、そのヒストグラムを作成し、該ヒストグラムを比較することにより、データ特性の変化を検出している。該データ特性判別部では、判別対象として、文字列データ、画像データともに適応可能である。データ判別法としては、領域識別法や、数理統計的な性質抽出法等がある。また、領域識別法としては、直交変換、濃度分布、濃度勾配、パターンマッチング等、またはこれらの複数の組み合わせを用いる方法がある。数理統計的な性質抽出法としては、主成分分析、ヒストグラム、期待値、分散値、連続性、連結性等、またはこれらの複数の組み合わせを用いる方法がある。また、領域識別法と数理統計的な性質抽出法とを組み合わせてもよ

い。

【0067】

【発明の効果】以上、説明したように、この発明によれば、符号化手段によって、辞書手段に登録された情報に基づいて、入力されるデータ列を符号化するとともに、そのデータ列と符号とを辞書手段に登録する符号化過程において、データ特性変化検出手段によって、入力される符号化前のデータ列の特性変化を検出し、データ列のデータ特性に変化があった場合には、辞書リセット制御手段によって、辞書内容および符号化手段の動作をリセットするようにしたので、複数のデータ列特性が混在するデータ列やデータ列特性変化部を持つデータ列に対しても、適切な辞書を作成することができるとともに、適切な符号化を実行することができ、高圧縮率を得ることができるという利点が得られる。

【図面の簡単な説明】

【図1】 本発明の第1の実施例によるデータ圧縮・伸張装置の符号化部の構成を示すブロック図である。

【図2】 本発明の第1の実施例によるデータ圧縮・伸張装置の復号化部の構成を示すブロック図である。

【図3】 本第1の実施例のデータ計数部によって作成されるヒストグラムを示す概念図である。

【図4】 本第1の実施例のデータ計数部における動作を説明するための概念図である。

【図5】 本第1の実施例によるデータ圧縮・伸張装置の符号化の手順を示すフローチャートである。

【図6】 本第1の実施例によるデータ圧縮・伸張装置の復号化の手順を示すフローチャートである。

【図7】 本発明の第2の実施例によるデータ圧縮・伸張装置の符号化部の構成を示すブロック図である。

【図8】 本発明の第3の実施例によるデータ圧縮・伸

張装置の符号化部の構成を示すブロック図である。

【図9】 本第3の実施例のデータ計数部における動作を説明するための概念図である。

【図10】 従来のLZW方式による符号化の手順を示すフローチャートである。

【図11】 従来のLZW方式による辞書登録テーブルの一例を示す概念図である。

【図12】 従来のLZW方式による符号化の過程を示す概念図である。

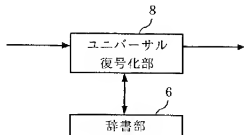
【図13】 従来のLZW方式による復号化の手順を示すフローチャートである。

【図14】 従来のLZW方式による復号化の過程を示す概念図である。

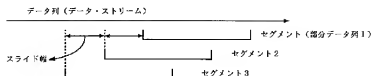
【符号の説明】

- 1 データ保持部
- 2 データ計数部（データ計数手段、データ特性変化検出手段）
- 3 比較部（データ特性変化検出手段、比較手段）
- 4 信号発生部（辞書リセット制御手段）
- 5 ユニバーサル符号化部（符号化手段）
- 6 辞書部（辞書手段）
- 7 辞書計数部（データ特性変化検出手段、辞書計数手段）
- 8 ユニバーサル復号化部
- 10 履歴計数部（履歴計数手段）
- 11 比較部（データ特性変化検出手段、比較手段）
- 12 データ計数部（データ特性変化検出手段、データ計数手段）
- 13 抽出部（データ特性変化検出手段、抽出手段）
- S1 辞書リセット制御信号
- S2 辞書リセット回避信号

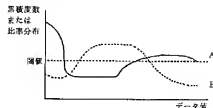
【図2】



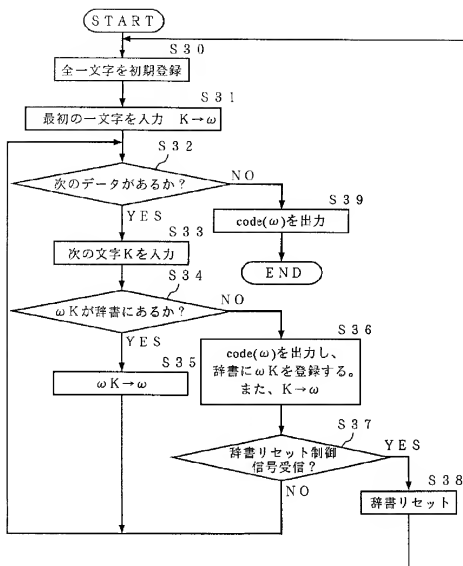
【図4】



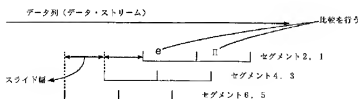
【図3】



【図5】



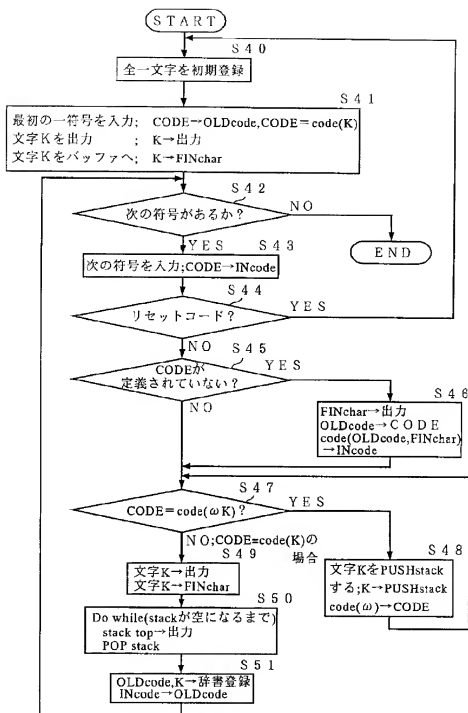
【図9】



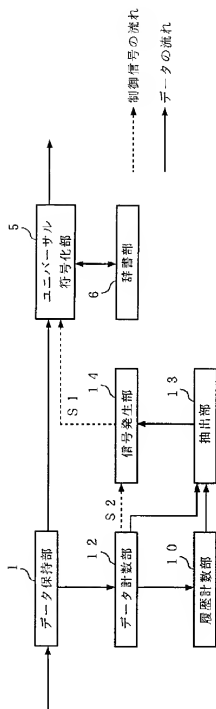
【図11】

辞書登録テーブル		
部分データ列 (string)	部分データ列 (ω K)	番号
a	a	1
b	b	2
c	c	3
a b	2 a	4
b a	2 a	5
a b c	4 c	6
c d	3 b	7
b a b	5 b	8
b a b a	8 a	9
a a	1 a	10
a a a	10 a	11
a a a a	11 a	12

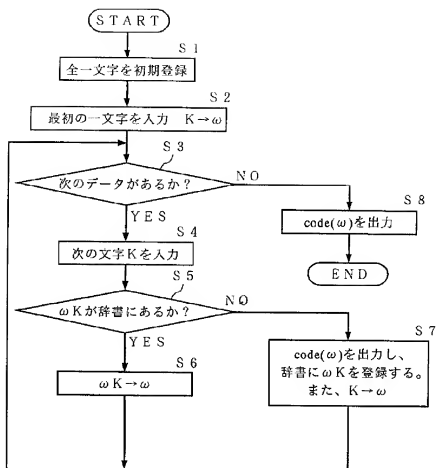
【図6】



【図8】



【図10】



【図12】



【図13】

